Principles of Secure System Design for Space Applications

Dr. Basel Halak
http://personal.southampton.ac.uk/bh1m10/

Learning Outcomes

UNIVERSITY OF Southampton
School of Electronics and Computer Science

At the end of this unit, you should be able to:

1. Explain the principles of Secure System Design Flow
2. Explain how Hardware Security Modules (HSM) and Trusted Platform Modules (TPM) technologies can be used to build robust defenses.
3. Discuss a case study of a secure-by-design product.

UNIVERSITY OF Southampton
School of Electronics and Computer Science

Threat Modelling Overview

- **What** is Threat modelling:

  It is process to identify, enumerate and prioritize potential threats, such analysis is typically done from a hypothetical attacker's point of view. It helps inform product developers on the required level of security defence mechanisms that need to implemented.

- **When** does it happen?

  This should take place in the early stages of product development cycle, and should be revisited regularly to keep up with emerging threats

- **What** is the expected output?

  Detailed documentation of the potential threats

UNIVERSITY OF Southampton
School of Electronics and Computer Science

Threat Modelling: **Why** do we need it?

A systematic approach to threat modeling brings about many benefits, which include:

1. It allows early identification of potential security problems
2. It helps develop a better understanding of security requirements
3. It allows for avoiding blind applications of unnecessary countermeasures
4. It provides a mechanism to identify new threats and adapt the design accordingly
5. Ultimately, develop cheaper and more secure products

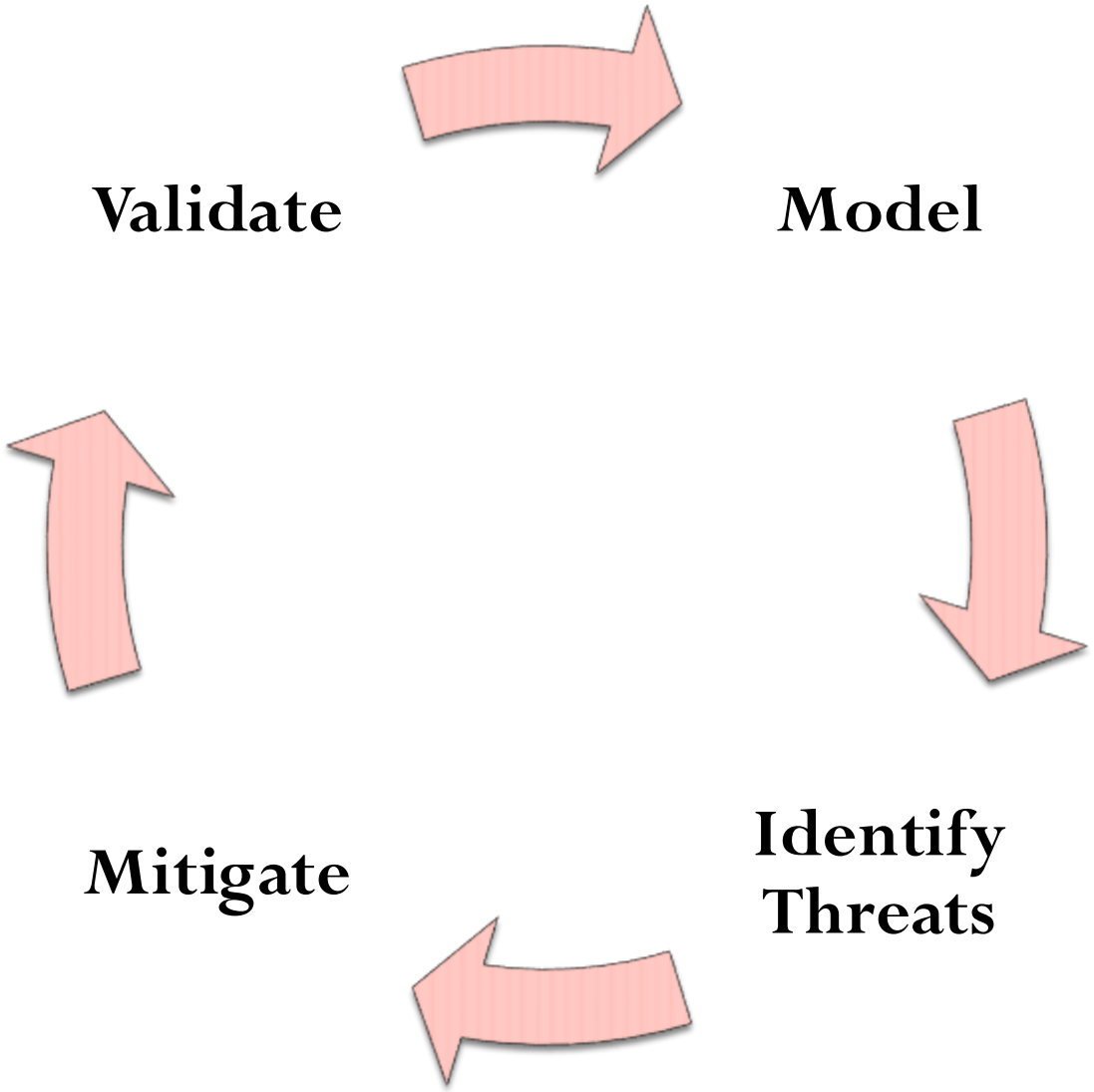
UNIVERSITY OF Southampton
School of Electronics and Computer Science

Threat Modelling: **How** to do it?

- To be able to perform effective security analysis, we need to answer the following questions:

1. What is the functionality/ architecture of the system you are building?
2. What part of our system do you care most about?
3. What are the possible attack scenarios?

Secure Design Cycle

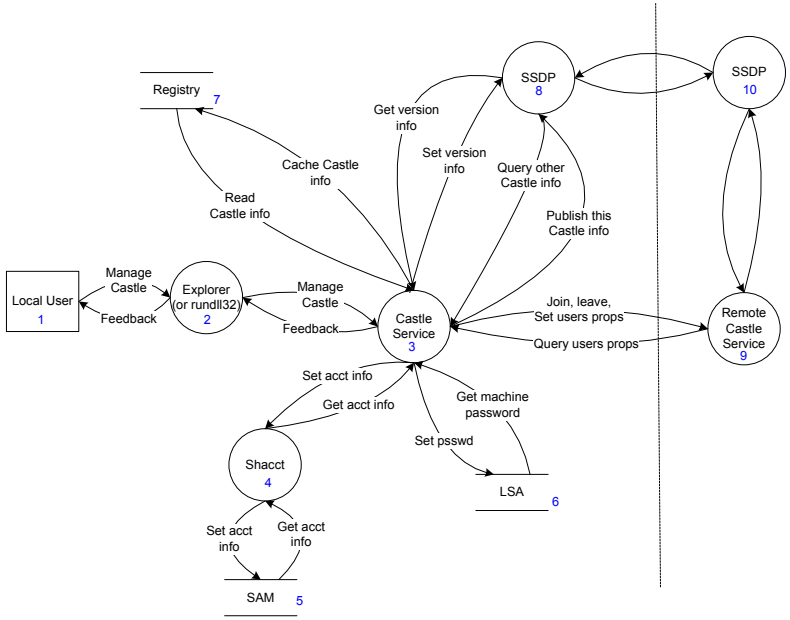


Before you start...

- Determine the key threat scenarios that are applicable to your product (e.g., understand the environment the product is going to operate in, and the security assumptions made on such an environment)
- Develop several use cases
- Define external entities and verify any assumptions made about them
- Identify assets and security objectives you want to achieve

Secure Design Cycle: Model

- **The purpose** of this step is to develop a detailed understanding of the system's operation and its components.
- One of the main approaches to achieve this is to use a data flow diagram (DFD)
- DFD is a graphical representation that provides information on how data, enters or leaves each system component, it also shows all system processes and identifies trust boundaries



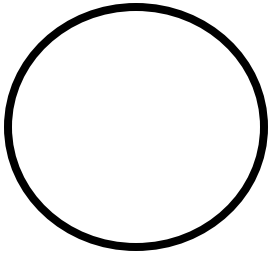
UNIVERSITY OF Southampton
School of Electronics and Computer Science

Elements of Data Flow Diagram

External Entity

Data Flow

Trust Boundary



Process

Data Storage

UNIVERSITY OF Southampton
School of Electronics and Computer Science

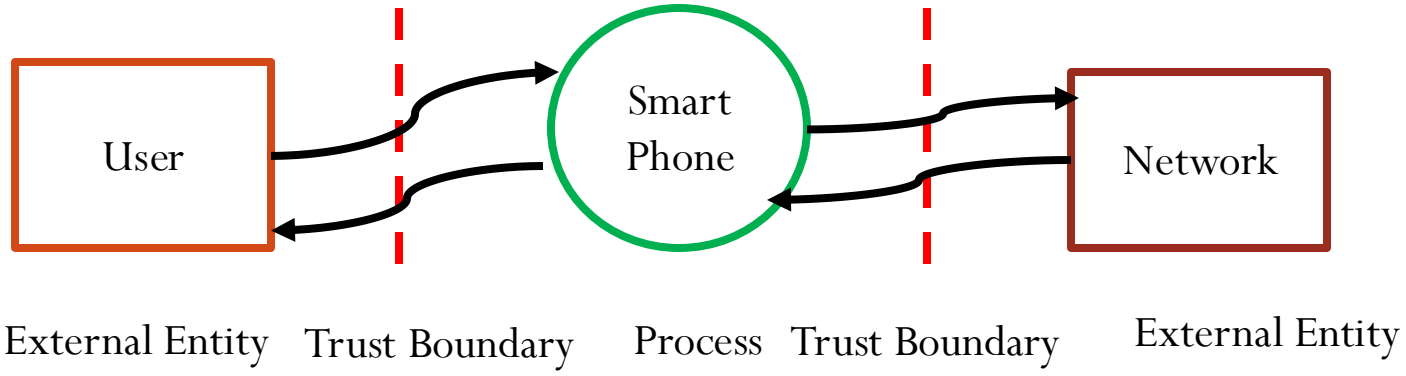
Elements of Data Flow Diagram

- **The process** (function, transformation) is part of a system that transforms inputs to outputs
- **Data flow** indicates the transfer of information (sometimes also material) from one part of the system to another
- **External Entity** are objects outside the system, with which the system communicates (e.g. sources and destinations of the system's inputs and outputs)
- **Data Storage** may include shared memory, storage device etc…
- **A Trust Boundary** typically intersect data flows and comprises points/surfaces where an attacker can interject. Examples include machine boundaries and privilege boundaries

UNIVERSITY OF Southampton
School of Electronics and Computer Science

Secure Design Cycle: Model

- **Example:** Model of a smartphone device



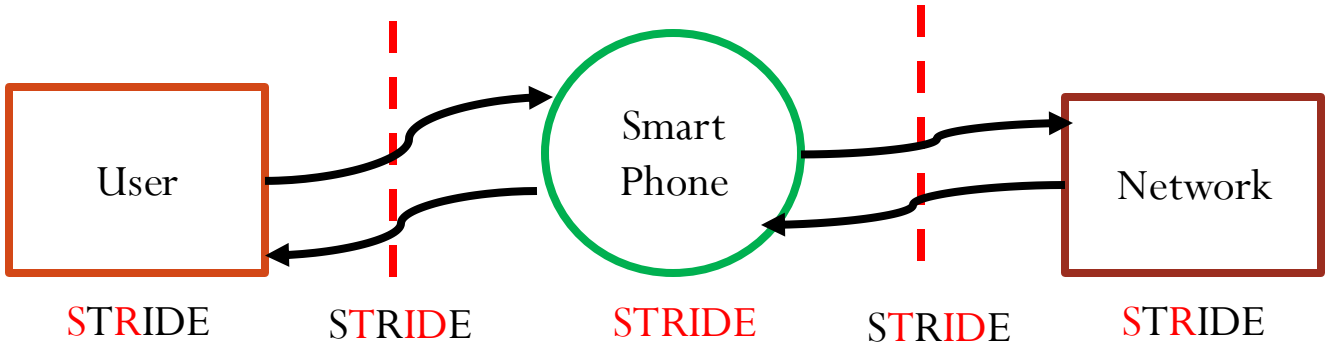
Secure Design Cycle: Identify Threat

UNIVERSITY OF Southampton
School of Electronics and Computer Science

Threat	Property	Definition	Example
Spoofing	Authentication	Impersonating something or someone else.	A virus pretending to be a system update
Tampering	Integrity	Modifying data or code	Modifying a data packet as it traverses the network
Repudiation	Non-repudiation	Claiming to have not performed an action	Denying a purchase
Information Disclosure	Confidentiality	Exposing information to someone not authorized to see it	Stealing secret information from a device
Denial of Service	Availability	Deny or degrade service to users	Crashing the web site, sending a packet and absorbing seconds of CPU time, or routing packets into a black hole
Elevation of Privilege	Authorization	Gain capabilities without proper authorization	Allowing a remote internet user to run commands is the classic example, but running kernel code from lower trust levels is also EoP

Secure Design Cycle: Identify Threat

- **Example:**



*Identified threats are in red color

STRIDE Security Objectives

UNIVERSITY OF Southampton
School of Electronics and Computer Science

- **Confidentiality**
  - Keeping data and resources secret or hidden
  - Conceal the existence of data
- **Integrity**
  - Refers to correctness and trustworthiness
  - Ensuring authorized modifications;
  - May refer to
    - Data integrity
    - Origin integrity (Authentication)
- **Availability**
  - Ensuring authorized access to data and resources when *desired*
    - *Often assume a statistical model for the pattern of use – which can be distorted*
- **Non-repudiation:** The party who sent and handled the data should be unable to deny the transactions it has made afterward.
- **Authentication:** All parties involved in a network should be authorized and able to identify each other.

UNIVERSITY OF Southampton
School of Electronics and Computer Science

Secure Design Cycle: Mitigate

- **One of the following mitigation approaches is applied to each threat identified in step 2:**
  1. Use standard security defences and countermeasures
  2. Develop new defence mechanisms (hard and takes a long time)
  3. Re-design your product to eliminate the vulnerability
  4. Ignore the threat …

UNIVERSITY OF Southampton
School of Electronics and Computer Science

Standard Defense Mechanisms-1

Threat	Desired Security Property	Typical Mitigation Techniques
Spoofing	Authentication	• User authentication approaches (e.g. biometric authentication, password-based ..) • Data/Code authentication using digital signatures or message authentication codes (MAC)
Tampering	Integrity	• Digital signatures • Message Authentication Codes
Repudiation	Non-Repudiation	• Digital Signatures

UNIVERSITY OF Southampton
School of Electronics and Computer Science

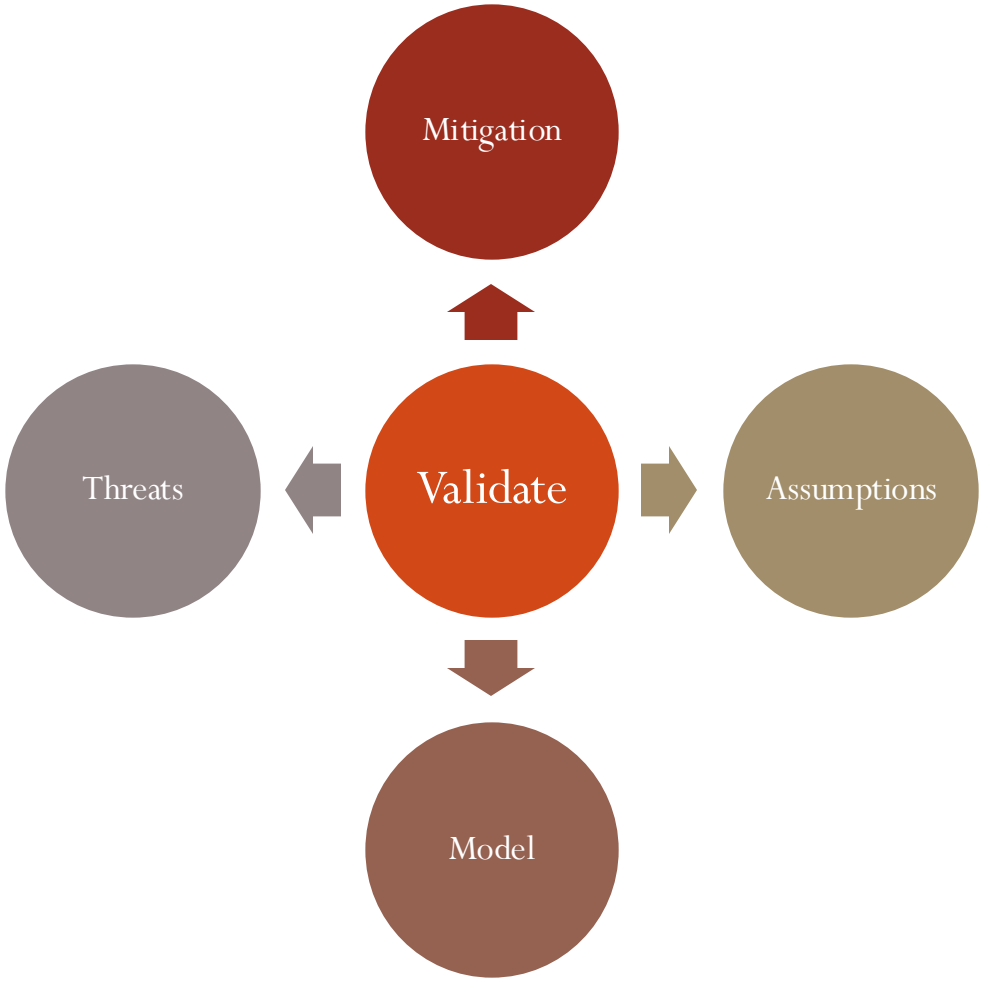
Standard Defense Mechanisms-2

Threat	Desired Security Property	Typical Mitigation Techniques
Information Disclosure	Confidentiality	Encryption
Denial of Service	Availability	• ACLs* • High availability designs (e.g., by increasing redundancy)
Elevation of Privilege	Authorization	• Monitor network traffic and behavior. • Institute a strong password policy • Multi-factor authentication

*ACL(Access Control List) is a network filter utilized by routers and some switches to permit and restrict data flows into and out of network interfaces. When an ACL is configured on an interface, the network device analyses data passing through the interface, compares it to the criteria described in the ACL, and either permits the data to flow or prohibits it.

UNIVERSITY OF Southampton
School of Electronics and Computer Science

Secure Design Cycle: Validate



Learning Outcomes

UNIVERSITY OF Southampton
School of Electronics and Computer Science

At the end of this unit, you should be able to:

1. Explain the principles of Secure System Design Flow
2. Explain how Hardware Security Modules (HSM) and Trusted Platform Modules (TPM) technologies can be used to build robust defenses.
3. Discuss a case study of a secure-by-design product.

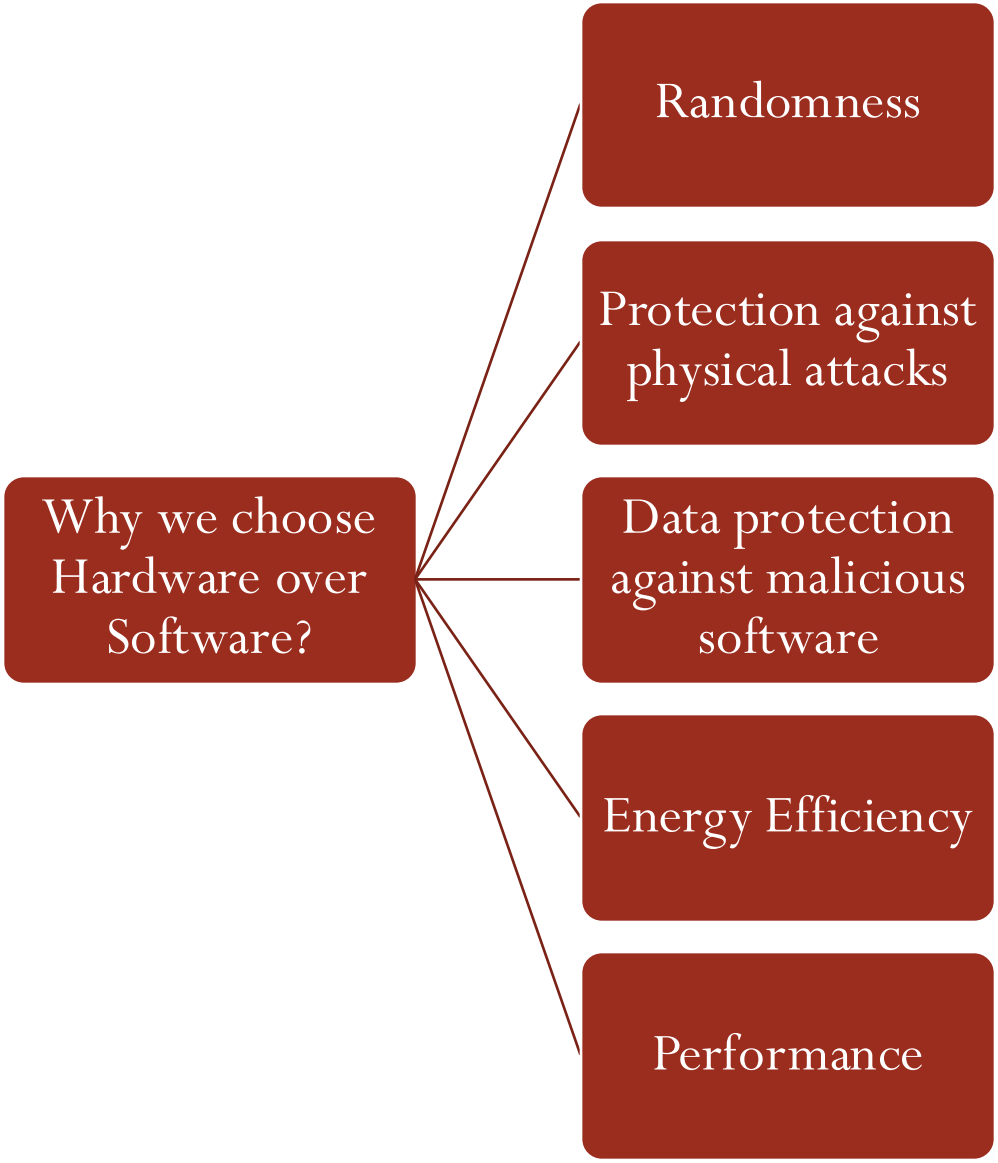
UNIVERSITY OF Southampton
School of Electronics and Computer Science

How to Develop a Secure System?

Hardware or Software Protection

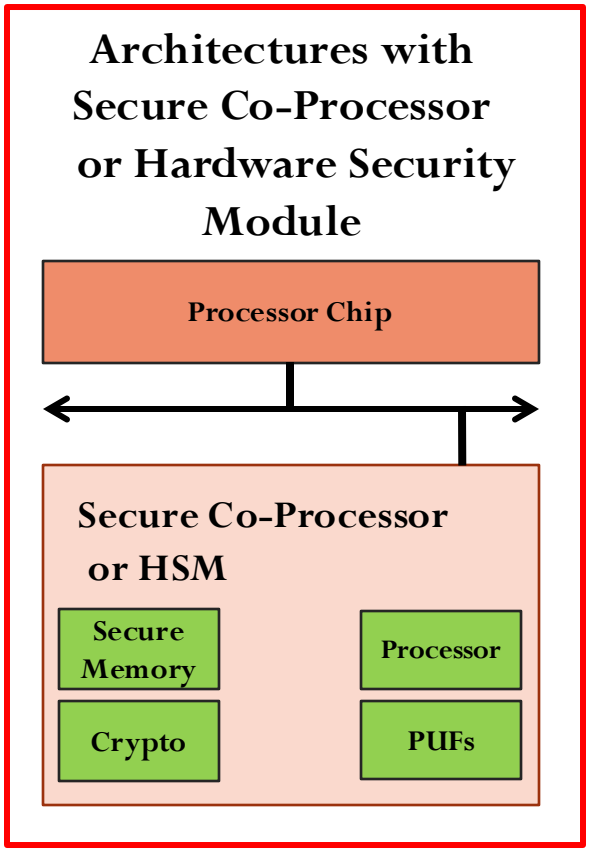
How to Develop a Secure System?

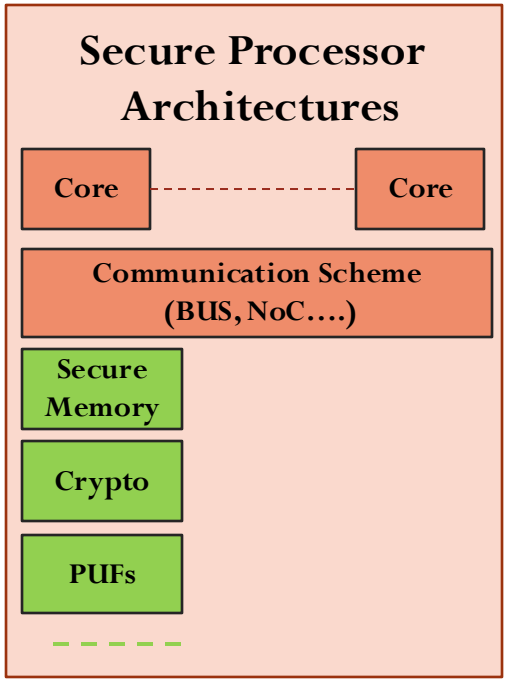
UNIVERSITY OF Southampton
School of Electronics and Computer Science

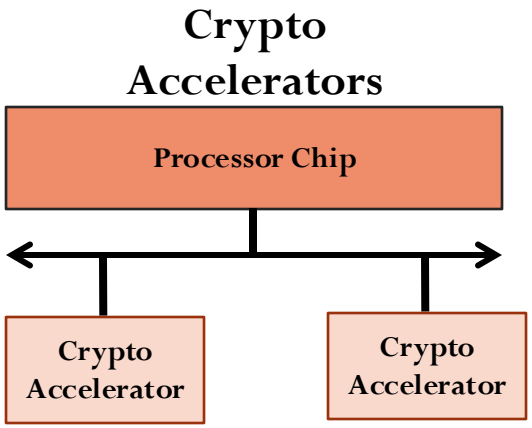


UNIVERSITY OF Southampton
School of Electronics and Computer Science

Types of Security-Related Architectures







UNIVERSITY OF Southampton
School of Electronics and Computer Science

Architectures with Secure Co-Processor or Hardware Security Module

- **Principle**: in this case a dedicated device is added to the system to perform cryptographic operations or run secure code, this can be attached to the system bus (PCIe) or or it can be integrated into an SoC design
- Examples include IBM card, and Intel-TPM (see on the side)

Architectures with Secure Co-Processor-1 Trusted Platform Module(TPM)

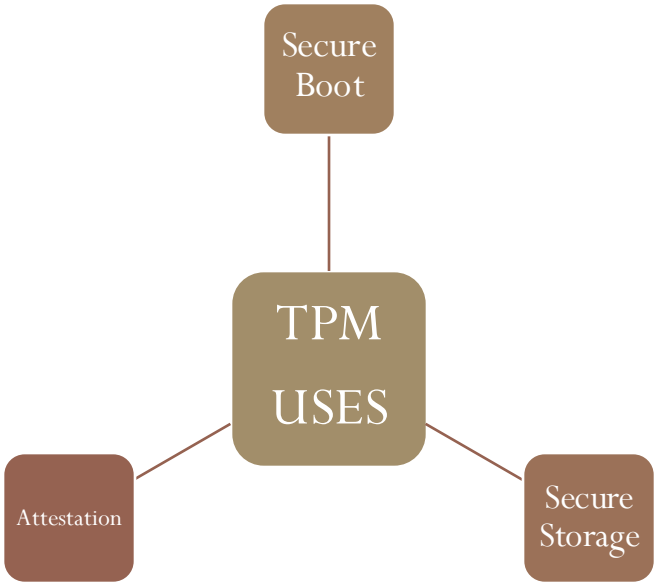
UNIVERSITY OF Southampton
School of Electronics and Computer Science

- **TPM specifications have been developed by the trusted computing group (TGC) to provide the following:**
  - Root of trust for measurement (RTM): to perform a trustworthy measurement of the state of the system, which includes its firmware and software elements that determines its operational state
  - Root of trust for storage (RTS): to provide an anti-tamper storage for secret cryptographic keys (SKR)
  - Root of trust for reporting (RTR) to provide anti-tamper storage for the device' s unique digital identifier i.e.. the endorsement key (EK), which is subsequently used to report the status of the system to third parties

UNIVERSITY OF Southampton
School of Electronics and Computer Science

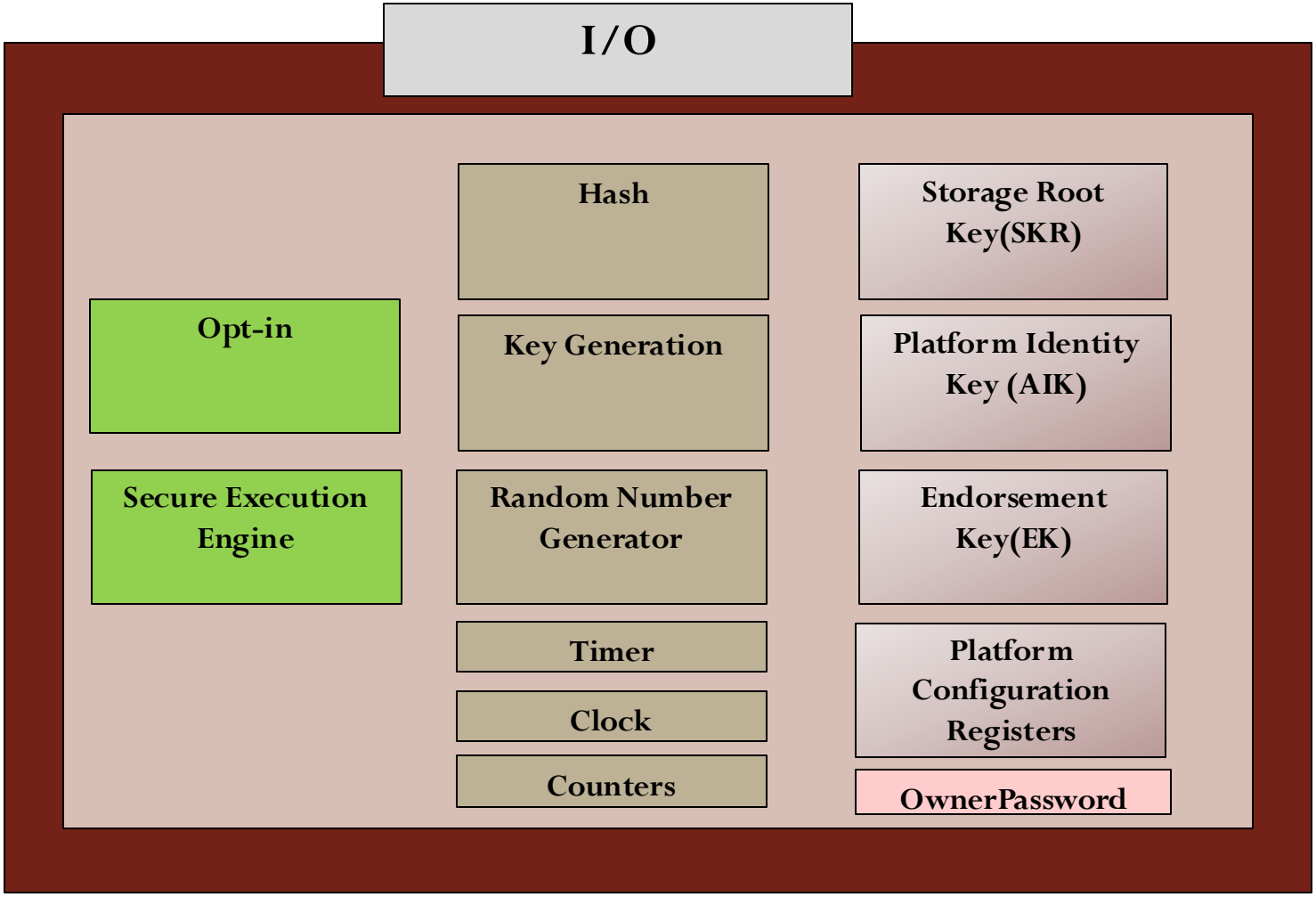
TPM Uses

- The main uses of the trusted platform modules are as follows:
  1) It provides a secure storage environment for sensitive information such as cryptographic keys.
  2) It allows detection of software tempering or any unwanted changes of the machine's configurations (i.e. secure boot)
  3) It is used to prove the integrity of the software running on the machine to other machines (e.g.. a server on the network), this is essential for attestation capabilities.



UNIVERSITY OF Southampton
School of Electronics and Computer Science

Trusted Platform Module



Secure Boot Principles

UNIVERSITY OF Southampton
School of Electronics and Computer Science

- **Secure Boot is a feature used to allow only cryptographically verified software to run on a device.**
- The motivation for secure boot is twofold:
  - **Costumer Perspective**:
    - Ensures the trustworthiness of the computing device by detecting any tampering of the software
  - **Manufacturer perspective:**
    - Ensures that customers only use their devices for the intended purposes
    - Ensures that customers only used approved software (i.e.. licenced installation)
    - Protects their devices form malware

Secure Boot Process

UNIVERSITY OF Southampton
School of Electronics and Computer Science

Stage 1
At the manufacturing stage

Create a key pair for an asymmetric encryption algorithm

↓

The application image is hashed to generate an application image (AI) digest

↓

The application AI digest is digitally signed by the OME private key to produce a signature

↓

The OME public key is securely stored on the commuting device

Stage 2
At the operation field

The application digest and list corresponding signature are loaded into the computing device

↓

The integrity of the AI digest is verified using the OME public key

Stored on the device

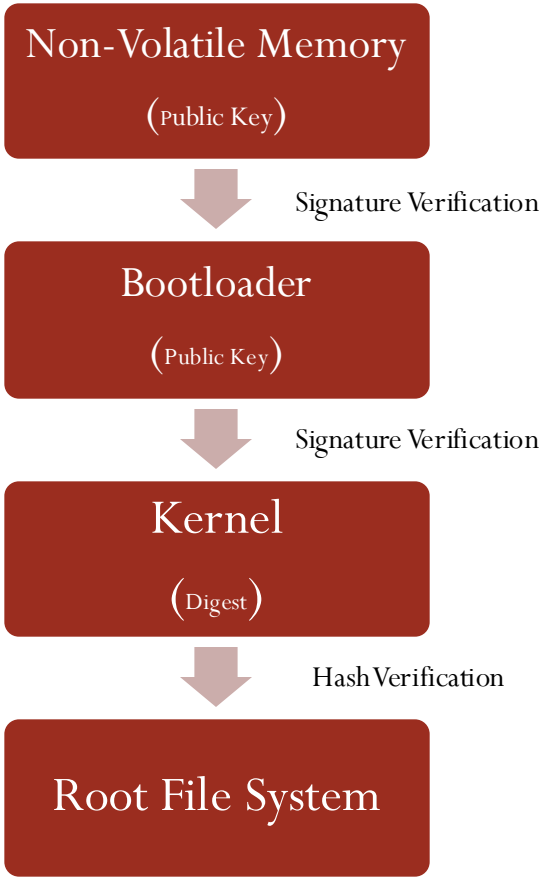
↓

The boot loader sequence can be executed or rejected based on the output of the verification

Secure Boot: The Chain of Trust

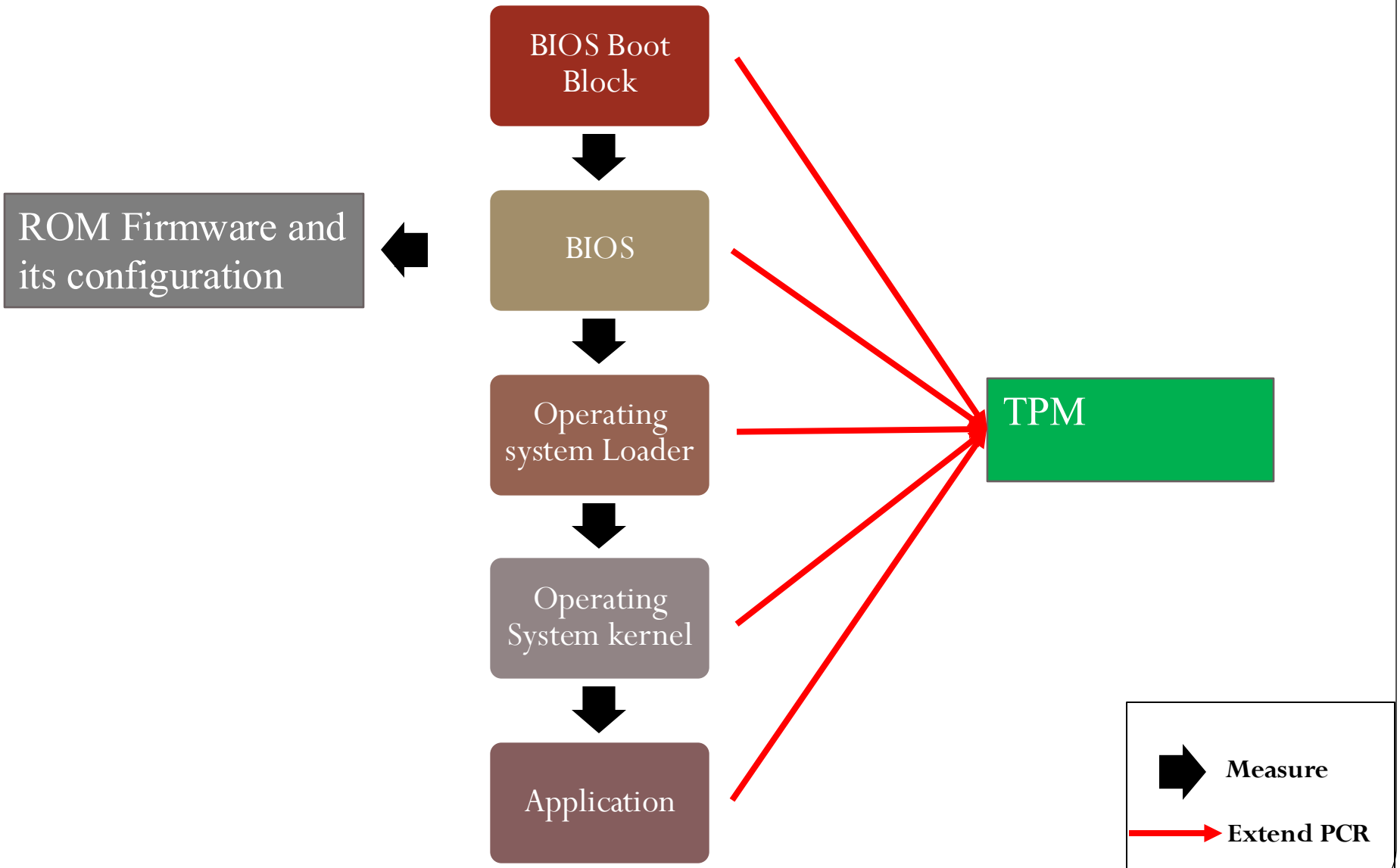
UNIVERSITY OF Southampton
School of Electronics and Computer Science

- The secure boot process is based on the idea of a chain of trust where each element authenticates the next using cryptographically-verified digital signature



TPM Secure Boot

UNIVERSITY OF Southampton
School of Electronics and Computer Science

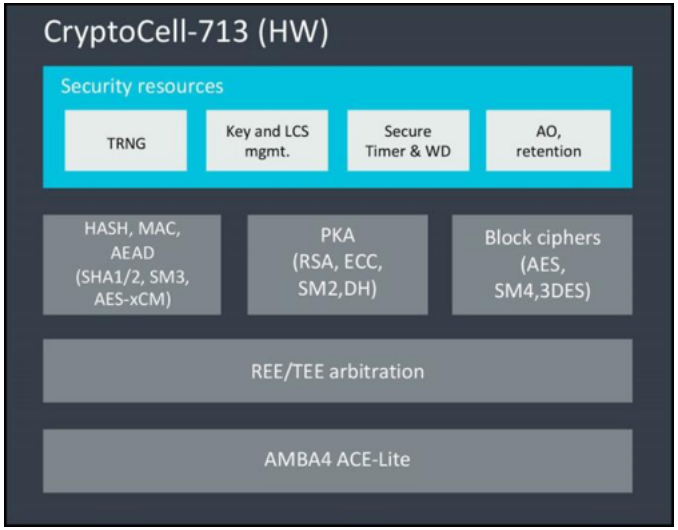


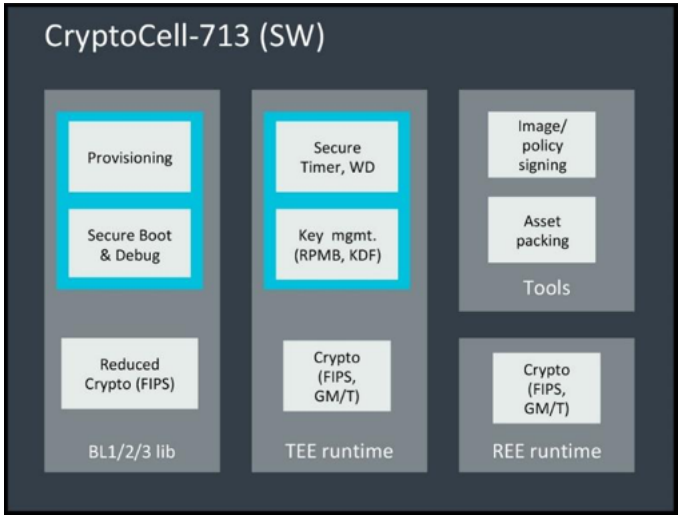
Architectures with Secure Co-Processor-2

ARM – CryptoCell

UNIVERSITY OF Southampton
School of Electronics and Computer Science

- Co-processor IP developed by Inside Secure
  - Model: CryptoCell-713
  - Co-processor attaching to ARM's buses
- Features:
  - crypto accelerators
  - side channel protection
  - supports Chinese crypto algorithms:
  - SM2 (public key alg. based on elliptic curves),
  - SM3 (hash function), and SM4 (block cipher)





https://community.arm.com/developer/ip-products/processors/b/processors-ip-blog/posts/new-cryptocell-security-ip-announced

Learning Outcomes

UNIVERSITY OF Southampton
School of Electronics and Computer Science

At the end of this unit, you should be able to:

1. Explain the principles of Secure System Design Flow
2. Explain how Hardware Security Modules (HSM) and Trusted Platform Modules (TPM) technologies can be used to build robust defenses.
3. Discuss a case study of a secure-by-design product.

Exercise: Threat Modeling a Space System

- **Scenario Overview**

You are performing a security analysis for a fictional Earth observation satellite system operated by "OrbitSafe Corp." Below is a high-level description of the system, which you will use to conduct the threat model.

- **System Description**
  - The satellite system is composed of four primary segments: User Segment, Ground Segment, Link Segment (Communication), and Space Segment (Satellite).
  - In the **User Segment**, mission operators and data analysts access satellite data and control the satellite through a software application called **OrbitView**. They authenticate via usernames, passwords, and multi-factor authentication (MFA) and have remote access to the platform.
  - The **Ground Segment** includes ground stations responsible for sending commands and receiving data, consisting of a Mission Control Center and a Data Processing Center. These stations use private networks, VPNs, and firewalls to secure data transfer and system integrity.
  - The **Link Segment** handles uplink and downlink communications, transmitting commands to the satellite and receiving data. This communication relies on radio frequencies and satellite-specific protocols, with encryption ensuring the security of data transfer.
  - In the **Space Segment**, the satellite captures high-resolution images, transmits data to the ground segment, and manages on-board operations through an autonomous computing system. Security measures include encrypted data storage, anti-jamming capabilities, and tamper-resistant hardware.
- **Task**: Each group will develop a threat model for the system. The aim is to analyze the description collaboratively, identify risks, and propose solutions.

What you need to do

UNIVERSITY OF Southampton
School of Electronics and Computer Science

- **Step 1: Identify Key Assets & Segments**
  Discuss and identify the critical assets in the system that need protection, considering all segments: user, ground, link, and space. Think about what data, controls, and interfaces are crucial to the system's functioning and security. For example, you might ask what data is most valuable, which communications are most sensitive, and where the main risks lie. As a deliverable, create a list of key assets and draw a high-level diagram of the satellite system, illustrating how segments connect and interact.
- **Step 2: Brainstorm Potential Threats**
  Using the STRIDE framework, identify threats for each segment. Consider spoofing attacks, where an adversary could impersonate legitimate users; tampering scenarios, where data or communications might be altered; and other concerns such as information disclosure risks, denial-of-service attacks, and privilege escalation attempts. For each identified threat, discuss its potential impact on system security and operations. The goal is to compile a list of threats for each segment and assess their possible implications.
- **Step 3: Analyze Risks & Prioritize**
  With a clear list of threats, analyze their likelihood and impact on the system. Discuss which threats are most probable and what consequences they might have if realized. Prioritize the risks based on severity and potential damage to the mission. As an outcome, produce a prioritized list of threats, emphasizing those that are both likely and have high potential impact.
- **Step 4: Propose Mitigations & Countermeasures**
  For the high-priority threats, brainstorm and propose mitigation strategies. Discuss how you might apply security controls, such as encryption, access control mechanisms, or tamper-resistant hardware. Consider the use of TPM to protect against specific threats. For example, you could debate how encryption might protect the data stream or whether tamper-resistant features could safeguard on-board systems. Your deliverable is a set of countermeasures for each high-priority threat, along with explanations of how these security measures address the identified risks.
- **Step 5: Report back your main findings**

Summary

- Security can never be an afterthought, so a systematic approach that considers it from the early stages of development is vital.
- Threat modelling is an essential tool to identify and mitigate security threats effectively with the minimum possible cost.
- The STRIDE model is widely used in this context, but you can also develop your own threat categories.